



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

JAMES

Atty. Ref.: 36-1116

Serial No.: 09/029,581

Group Art Unit: 2171

Filed: March 6, 1998

Examiner: Al Hashemi, S.

For: DATABASE ACCESS

5C
#28
4/6/04

March 22, 2004

RECEIVED

SECOND APPEAL BRIEF

MAR 30 2004

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Technology Center 2100

Sir:

Appellant hereby appeals the Final Rejection of October 22, 2003, Paper No. 26.

REAL PARTY IN INTEREST

The real party in interest is the assignee, British Telecommunications plc, a corporation of the country of Great Britain.

RELATED APPEALS AND INTERFERENCES

The Appellant, the undersigned, and the assignee are not aware of any related appeals or interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

03/26/2004 SLUANG1 00000029 09029581

02 FC:1402

330.00 OP

JAMES
Application No. 09/029,581
March 22, 2004

STATUS OF THE CLAIMS

Claims 1-9 remain pending in this application. Claims 1-8 stand rejected by the Examiner, the rejections of which are appealed. No claims have been substantively allowed.

Claim 9 was not substantively addressed in the Final Rejection. That is, no specific ground of rejection was provided for claim 9 in the Final Rejection.

STATUS OF ANY AMENDMENT FILED SUBSEQUENT TO FINAL REJECTION

No Amendment has been filed subsequent to the Final Rejection. The claims as presented in the appendix to this brief are as amended by the March 6, 1998 Preliminary Amendment. While claim 9 was not specifically rejected, claim 9 has been presented in the appendix to this brief.

CONCISE EXPLANATION OF THE INVENTION

The present invention relates to a client/server computer environment in which data consistency between data held in a master database and data held in a cache database is checked. In particular, the data consistency is checked by comparing a key associated with a respective cache database entry and a key associated with an index to a corresponding data entry in the master database. An exemplary embodiment of the present invention is shown in Figs. 1, 3c and 6. The structure shown in Figs. 3a and 3b and processes shown in Figs. 4 and 5 illustrate acknowledged prior art systems which can be read in contrast to Figs. 3c and 6 to understand the exemplary embodiment of the present invention.

The computer environment in accordance with an exemplary embodiment of the present invention includes, inter alia, a file server 100 which runs network and database management system (DBMS) software suitable for providing remote database access to a plurality of clients 130 over a network 140. The file server 100 includes a main memory 104 which is split into areas for standard program and data storage. The main memory 104 includes a large area of memory known as a system global area (SGA) 105 which can be used by the database software to speed up access to a master database 126 for the clients 130. (See Fig. 1 and page 7, lines 30-36).

The clients 130 are standard computer systems which run software suitable for reading and writing data stored in a database. The clients 130 each include a standard disk drive 135 having a storage area for a cache database 136 into which data from the master database 126 can be copied and accessed. (See Fig. 1 and page 8, lines 1-8).

The master database 126 is operatively coupled to the file server 100 and includes a data table 204 for storing a group of related data. The data table 204 is split into a plurality of pages 215 each comprising rows of data 210. The master database 126 also includes an index 206 comprising index pages 220. The index pages 220 contain index entries 225 for each row 210 in the data table 204. Each index entry 225 typically comprises index fields, one of which references a specific row of the data pages 215. (See Fig. 2 and page 8, lines 9-27).

The cache database 136 comprises a storage area 230 for holding cached data. The cached data typically comprises data rows 235 which are copied from the master database 126. The cache database 136 may or may not have a corresponding index depending on its size and acceptable access speed. (See Fig. 2 and page 9, lines 14-22).

Fig. 5 describing the prior art and Fig. 6 describing the present invention illustrate database read procedures for systems operating without and with a key in the index entries, respectively. In the procedure illustrated by prior art Fig. 5, a client 130 transmits a query across the network 140 to the file server 100 to read data from the file server 100 (step 500). The query includes an identifier for the data row requested and a respective key. The file server 100 ultimately compares the key of a cached data row 235 with the key of a master data row 210 which has been requested (step 525). If the keys are the same, the file server 100 sends a reply to the client 130 that the cached data is still valid (step 540). If, however, the keys are different, the file server 100 transmits the entire data row 210 (or only the requested columns of the data row 210) to the client 130 to update the cache database 136 (step 535). The client 130 receives the response and acts accordingly (step 545). (See page 11, line 19 to page 12, line 4).

In the procedure of Fig. 6 illustrating the present invention, steps 600 to 620 are equivalent to steps 500 to 520 of prior art Fig. 5. However, in step 625, the file server 100 compares the key in the index entry of the requested data row 210 with the key of the cached data row 235 as received in the query. Thus, the key of an index entry is read and used in the comparison in step 625 of the present invention rather than the key of a row from the master database 126 as in prior art Fig. 5. If the keys are the same, the file server 100 transmits a message to the client 130 that the cached data is still valid (step 640). If, however, the keys are different, the file server 100 accesses the master database 126 to retrieve the current data row (step 630). The current data row is then returned to the client 130 to update the cache database 136 (step 635). (See page 12, lines 5-15).

The advantage of reading the key of an index entry and using it in the comparison in step 625 of the present invention (rather than the key of a row from the master database 126 as illustrated in step 525 of prior art Fig. 5) is that it avoids the need to read the whole row from the master database 126 and thus reduces the processing load on the file server 100. (See page 12, line 19 to page 13, line 6.)

Although the specification of the present application describes the use of time stamps as an example of the “keys” discussed above, other methods for marking data rows to enable data consistency comparisons are possible such as an incremental counter field or unique codes representative of the specific state of a row. (See page 13, lines 24-34.)

CONCISE EXPLANATION OF THE ISSUE PRESENTED FOR REVIEW

Whether claims 1-8 are patentable under 35 U.S.C. §102(e) as not having been anticipated by Micka et al (U.S. Patent No. 5,592,618).¹

WHETHER THE CLAIMS STAND OR FALL TOGETHER

Claims 1 and 4 stand or fall together and do not stand or fall with any other claims.

Claims 2 and 8 stand or fall together and do not stand or fall with any other claims.

Claim 3 stands or falls alone and does not stand or fall with any other claims.

Claim 5 stands or falls alone and does not stand or fall with any other claims.

Claim 6 stands or falls alone and does not stand or fall with any other claims.

Claim 7 stands or falls alone and does not stand or fall with any other claims.

¹ As noted above, claim 9 was not specifically rejected.

ARGUMENTS WITH RESPECT TO THE ISSUES PRESENTED FOR REVIEW

Claims 1-8 are patentable under 35 U.S.C. §102(e) as not having been anticipated by Micka et al (U.S. '618, hereinafter "Micka").

For a reference to anticipate a claim, each element must be found, either expressly or under principles of inherency, in the reference. Micka fails to disclose each element of the claimed invention. For example, Micka fails to disclose the following limitation of independent claim 1:

“...comparing a first key stored in association with the item of data in the cache database with **a second key stored in association with an index entry** for the respective item of data in the master database (emphasis added).”

Similarly, Micka fails to disclose the following limitations of independent claim 2:

“reading a first key stored in association with a cached copy of a required item of data from the cache database;
reading a **second key stored in association with an index entry** for a respective item of master data from the master database;
comparing the first key with the second key...(emphasis added).”

Claims 1 and 2 each requires comparing a first key stored in associated with an item of data in a cache database with a second key stored in association with an **index entry** for a respective item of data in a master database.

Micka discloses a master journal 800 and state table 700 for enabling secondary host 411 to store data received from primary host 401 in the correct order and to enable secondary host 411 to operate on a stand-alone basis if primary host 401 ceases to exist. (*See, e.g.,* col. 11, lines 1-5). However, Micka fails to disclose comparing a first key stored in association with an item

JAMES
Application No. 09/029,581
March 22, 2004

of data in a cache database with a second key stored in association with an index entry for a respective item of data in a master database.

The Final Rejection apparently alleges that col. 9, lines 63-67, col. 10, lines 1-25 and/or steps 710 and 712 in Fig. 4 of Micka disclose this claimed feature. (See sections 1 and 2 of the Final Rejection). Appellant respectfully disagrees. Col. 9, line 63 to col. 10, line 25 is reproduced below:

“Referring now to FIG. 3, the record set information 600 is generated by the primary storage controllers 405 and collected by the PDM 404. Update Specific Information 601-610, includes a primary device unit address 601 of each record set indicating the actual primary DASD 406 that the record update occurred on. A cylinder number/head number (CCHH) 602 indicates a location on primary DASD 406 for each record set. Primary SSID 603, the primary storage controller session identifier is the same as primary SSID 505. Status flags 604 provide status information regarding whether specific data records 620 follow. Sequence number 605 assigns a number to each record set for indicating whether the entire record set has been read (all data transferred to the PDM 404) in conjunction with a pseudo count field 630. Primary DASD write I/O type 606 is an operation indicator identifying the type of write operation performed on each record in the record set, the operation indicators including: update write; format write; partial track records follow; full track data follows; erase command performed; or write any performed. Search argument 607 indicates initial positioning information for the first read record set data record 620. A sector number 608 identifies that sector that the record was updated at. Count field 609 describes a number of specific record count, key and data fields 620 that follow. A host application time when the primary DASD 406 write update occurred is recorded in time of updates 610. Specific record data 620 provides a count/key/data (CKD) field of each record update. Lastly, the pseudo count field 630 is compared to a predetermined pattern, for example eight bytes of “FF”, indicating whether the entire read record set was transferred to the PDM 404.”

No portion of the above passage discloses comparing a first key stored in association with an item of data in a cache database with a second key stored in association with an index entry

JAMES
Application No. 09/029,581
March 22, 2004

for a respective item of data in a master database. The above passage of Micka merely describes record set information 600 which, like state table 700 and master journal 800, is transmitted from primary host 401 to secondary host 411 to enable secondary host 411 to know that it has successfully received all of the data transmitted to it and how and where to store the received data so that secondary host 411 can successfully and exactly replicate what is stored at primary host 401. Items 710 and 712 of Fig. 1 are simply parts of state table 700. As discussed above, state table 700 is used by Micka to help the secondary host 411 know what data it should have received from primary host 401 and where and how to store it. Items 710 and 712 are never compared with a key stored in association with an index entry for data in a master database.

Master journal 800 disclosed in Fig. 5 of Micka does not disclose the claimed index entry for a respective item of data in a master database, or a key stored in association with an index entry. Col. 11, lines 2-5 of Micka states, "The state table 700 and master journal 800 support disaster recovery, and hence must be able to operate in a stand-alone environment wherein the primary system 401 no longer exists." State table 700 and master journal 800 are thus not stored in the primary system 401. Indeed, col. 6, lines 30-31 of Micka states "FIG. 5 is a master journal as used by the secondary site of FIG. 1." State table 700 and/or master journal 800 therefore does not disclose an index for a respective data item in a master database or a key stored in association with the index entry or comparison of that key to another key stored in association with an item of data in a cache database.

While sections 1 and 2 of the Final Rejection apparently allege that col. 9, lines 63-67, col. 10, lines 1-25 and/or steps 710 and 712 in Fig. 4 of Micka discloses comparing a first key stored in association with an item in a cache database with a second key stored in association

JAMES
Application No. 09/029,581
March 22, 2004

with an index entry for a respective item of data in a master database, the “Response to Arguments” section of the Final Rejection apparently alleges that col. 14, lines 21-28 discloses this claimed feature. (Page 5, lines 6-10 of the Final Rejection). Appellant respectfully disagrees. Col. 14, lines 21-28 states the following:

“Copy validation presents several barriers including the potentially high cost of re-transmitting the copy back to the primary for comparison, the extra system time required to perform the copy audit, and comparing the secondary copy to the primary data at a given point in real time, when some primary data may have again been updated and is no longer equivalent to the secondary copy.”

While this portion of Micka discloses copy validation of data at a primary site which has been copied at a remote site, this portion of Micka fails to disclose an index entry for a respective item of data in a master database, let alone a key stored in association with the index entry. Indeed, the Final Rejection fails to specifically indicate which item of Micka discloses the claimed index entry for a respective item of data in a master database, let alone a key stored in association with the index entry. For example, Micka fails to disclose any index entry of data items in primary host 1200 or a key stored in association with an index entry for a data item in primary host 1200. If the Examiner maintains the rejection over Micka, Appellant respectfully requests that the Examiner specifically provide clarification regarding what feature of Micka discloses the (1) index entry for a respective item of data in a master database, (2) a key stored in association with the index entry, and (3) comparison of that key to another key stored in association with an item of data in a cache database. Col. 14, lines 21-28 of Micka does not disclose any of these claimed features as apparently alleged by the “Response to Arguments” section of the Final Rejection.

JAMES
Application No. 09/029,581
March 22, 2004

Claim 2 further requires “retrieving in the event the first and second keys are the same the cached copy of the item of data or in the event the first and second keys are different the respective item of master data.” Appellant respectfully submits that Micka fails to disclose this additional feature. (See page 4, lines 1-3 of the Final Rejection). The Final Rejection alleges that col. 13, lines 28-36 of Micka discloses this feature. Appellant respectfully disagrees. Col. 13, lines 28-36 of Micka states the following:

“If a time interval group is incomplete, then step 1110 retries reading the record sets from the primary storage controller 405 until the required data is received. If errors occur, a specific duplex volume pair or pairs may be failed. Having received complete time interval groups, step 1120 determines a first consistency group journal record. The first (or current) consistency group journal record is that record which contains the earliest time of update 610.”

This portion of Micka discloses groups of data being transferred from primary host 401 to secondary host 411. If a data group is incomplete, data from primary host 401 is reread so that it may be properly transferred from primary host 401 to secondary host 411. This portion of Micka has nothing to do with retrieving data from a cache database or a master database based on whether the compared first and second keys are the same or different. If anything, this portion of Micka highlights the difference between the present invention which relates to caching and the Micka system which relates to disaster recovery (dr).

In the normal operation of Micka’s system, if a client wished to retrieve a data item, the client would retrieve it automatically from primary host 1200. Only if a disaster prevents the primary host 1200 from providing the data item would the client instead make a direct request to the secondary data host 1210 for the desired data item. The client would thus never make a

JAMES
Application No. 09/029,581
March 22, 2004

single request expecting it to be fulfilled by either the primary or secondary data host 1200 or 1210 as determined automatically by the combination of the primary and secondary data hosts.

The “Response to Arguments” section of the Final Rejection states, *inter alia*, “...referring to Fig. 4, col. 10, lines 49-65, Micka discloses all the elements subject matter in claim 2.” (See page 5, lines 16-17 of the Final Rejection). Appellant disagrees with the apparent allegation that col. 10, lines 49-46 of Micka discloses all of the claimed features required by claim 2. Col. 10, lines 49-65 states the following:

FIGS. 4 and 5 show a state table 700 and a master journal 800, respectively, for describing a current journal contents, which simplifies recovery time and journal transfer time. The state table 700 provides configuration information, collected by and common to the PDM 404 and SDM 414, and includes primary storage controller session identifiers (SSID numbers) and the volumes therein, and the corresponding secondary storage controller session identifiers and the corresponding volumes. Thus the configuration information tracks which primary volumes 710 or primary DASD extends map to secondary volumes 711 or secondary DASD extends. With a simple extension to the state table 700 indicating partial volume extends 712 (CCHH to CCHH), partial volume remote copy can be accomplished using the same asynchronous remote copy methods described herein, but for a finer granularity (track or extent) than full volume.

Col. 10, lines 49-65 fails to disclose all of the claimed subject matter required by claim 2 as alleged by the Final Rejection. For example, col. 10, lines 49-65 fails to disclose comparing a first key with a second key stored in association with an index entry for a respective item of master data from a master database or retrieving in the event the first and second keys are the same, the cached copy of the data item. Fig. 4 of Micka merely shows state table 700. Col. 10, lines 49-65 merely describes how state table 700 stores a mapping between locations on the primary host store and corresponding locations on the secondary host store. There is no

JAMES
Application No. 09/029,581
March 22, 2004

disclosure of comparing keys as required by claim 2 or one of the keys being stored in association with an index entry for a respective item of data from a master database.

Claims 3 and 7 depend from claims 1 and 2, respectively. Claims 3 and 7 require the first and second keys being time-stamps. Accordingly, the invention required by claims 3 and 7 (read in conjunction with base claims 1 and 2, respectively) involve comparing first and second time-stamps, the first time stamp being stored in association with a cached copy of a required data item and the second time stamp being stored in association with an index entry of a data item from the master database. Micka fails to disclose this feature.

Independent claim 5 requires a database file server that includes means for accessing an index of the database and reading an index entry for requested data items, the index entry including a second key for the stored item of information. Once again, Appellant submits that Micka fails to disclose a key associated with an index entry. Micka further fails to disclose comparing a first key from a local cache with the second key from the index entry of a master database.

Claim 5 further requires means for returning an indication that the cached data item is consistent with the master database if the two compared keys are the same or returning a copy of the requested item of data from the master database if the two compared keys are different. There is no disclosure of this feature in Micka. The Final Rejection indicates that col. 10, lines 35-47 of Micka discloses this claimed feature. Appellant respectfully disagrees. Col. 10, lines 35-47 merely discloses determining whether all of the records in a primary host have been properly transferred to the secondary host. This portion of Micka has absolutely nothing to do

JAMES

Application No. 09/029,581

March 22, 2004

with returning an indication of data consistency or returning a copy of requested data from a master database depending on whether two compared keys are the same or different.

Claim 6 is directed to a database index having an index entry which includes version information which changes each time respective information in a database changes. The Final Rejection apparently alleges that col. 11, lines 6-11 of Micka discloses this claimed feature.

Appellant respectfully disagrees. Col. 11, lines 6-11 states the following:

“A time stamp control is placed at the front and back of each master journal 800 to ensure that the entire control entry was successfully written. The time stamp control is further written to the secondary DASDs 417. The control elements include dual entries (1) and (2), wherein one entry is always a current entry, for example: (1) Timestamp control /Control Info/ Timestamp Control....”

While this portion of Micka discloses a time stamp being placed on a master journal entry, this portion of Micka fails to disclose changeable version information of a database index which changes each time information in the database itself changes.

In view of the foregoing, it is respectfully submitted that claims 1-8 are patentable as not having been anticipated by Micka.

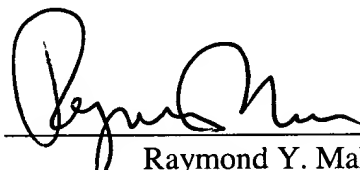
JAMES
Application No. 09/029,581
March 22, 2004

CONCLUSION

For all of the reasons set forth above, it is respectfully requested that this appeal be granted and that the rejections discussed above be reversed.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: 
Raymond Y. Mah
Reg. No. 41,426

RYM/sl
1100 North Glebe Road, 8th Floor
Arlington, VA 22201-4714
Telephone: (703)816-4000
Facsimile: (703)816-4100

APPENDIX OF CLAIMS ON APPEAL

1. A method for checking the consistency of an item of data in a cache database with a respective item of data in a master database by comparing a first key stored in association with the item of data in the cache database with a second key stored in association with an index entry for the respective item of data in the master database.

2. A method for retrieving an item of data from one of a cache or a master database, the master database comprising a plurality of items of master data and an index containing entries corresponding to one or more of the items of master data, the cache database containing a cached copy of at least one item of the master data, the method comprising the steps of:

reading a first key stored in association with a cached copy of a required item of data from the cache database;

reading a second key stored in association with an index entry for a respective item of master data from the master database;

comparing the first key with the second key; and

retrieving in the event the first and second keys are the same the cached copy of the item of data or in the event the first and second keys are different the respective item of master data.

3. A method according to claim 1, wherein the first and second keys are time-stamps.

4. Use of a method according to claim 1 in a client/server system.
5. A database fileserver apparatus comprising:

input means for receiving a conditional read request for an item of data stored in the database, the request including a first key for a previously-retrieved copy of the item of data;

means for accessing an index of the database and reading an index entry for the requested item of data, the index entry including a second key for the stored item of information;

means for comparing the first and second keys; and

means if the keys are the same for returning an indication that the previously-retrieved copy of the item of data is consistent or if the keys are different for reading from the database and returning a copy of the item of data.
6. A database index, wherein at least one index entry in the index includes at least:

identity information for identifying an item of information in the database;

location information for indicating the location in the database of the item of information;

and

version information which changes each time the respective information in the database changes.
7. A method according to claim 2, wherein the first and second keys are time stamps.

JAMES

Application No. 09/029,581

March 22, 2004

8. Use of method according to claim 2 in a client/server system.

9. A method for checking the consistency of an item of data in a cache database with a respective item of data in a master database by comparing a first key stored in association with the item of data in the cache database with a second key stored as a component of an index entry for the respective item of data in the master database.